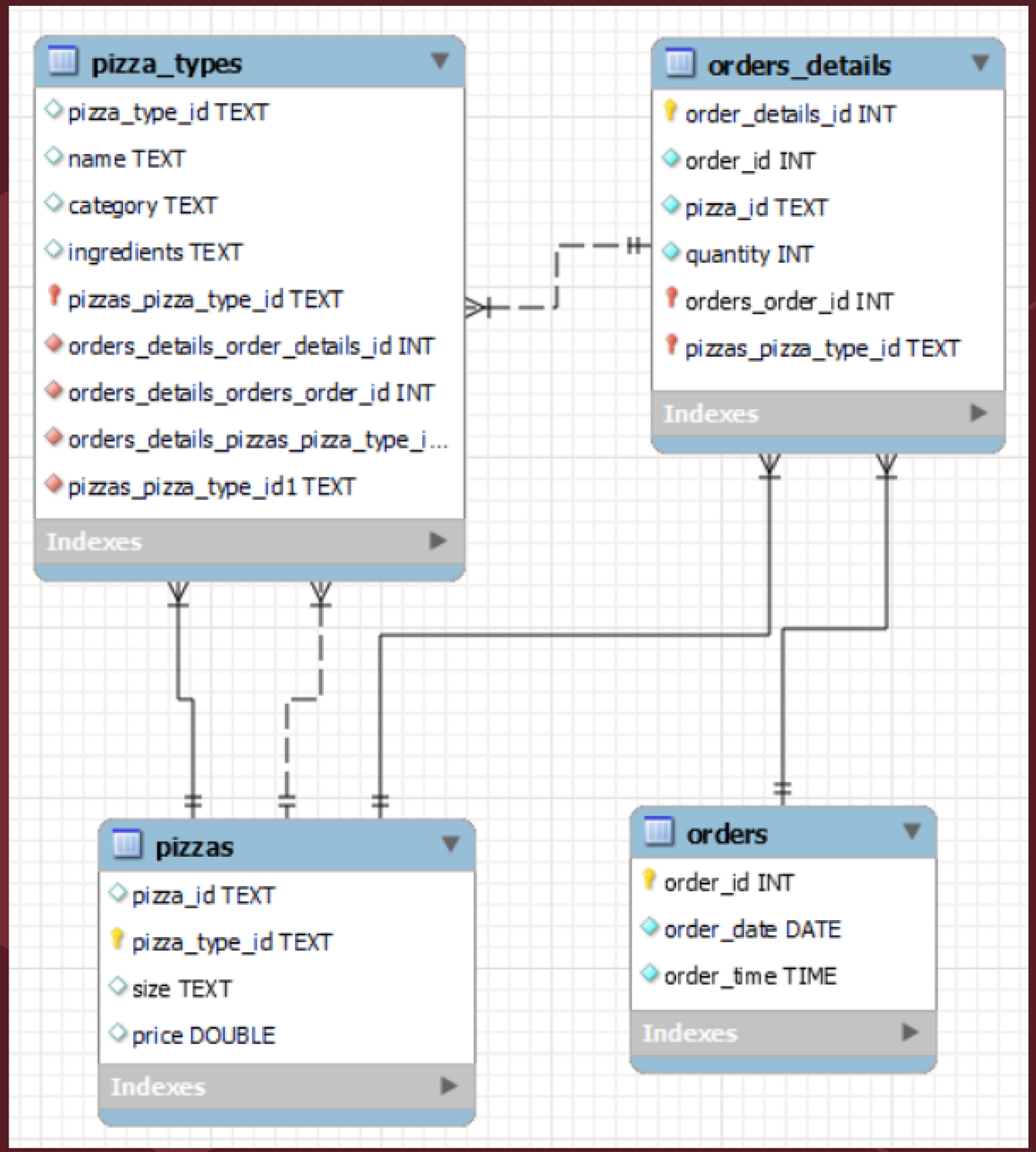
SALES REPORT

02 August, 2024

INTRODUCTION

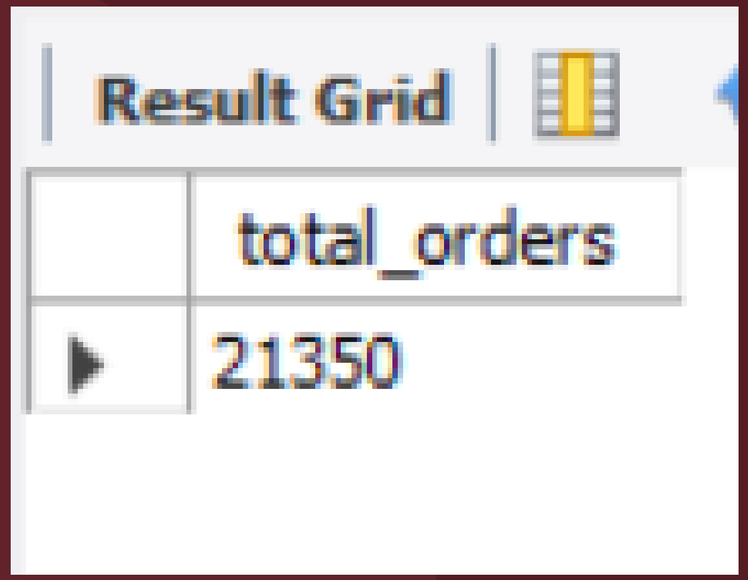
Welcome to our Sales Report Presentation. It includes an overview of our sales performance, exploring the highs, challenges, and strategic insights. In this project I used SQL queries to solve questions that were related to pizza sales.

ER Diagram


pizza_types
pizza_type_id TEXT
name TEXT
category TEXT
ingredients TEXT
pizzas_pizza_type_id TEXT
orders_details_order_details_id INT
orders_details_orders_order_id INT
orders_details_pizzas_pizza_type_j...
pizzas_pizza_type_id1 TEXT
Indexes
orders_details
order_details_id INT
order_id INT
pizza_id TEXT
quantity INT
orders_order_id INT
pizzas_pizza_type_id TEXT
Indexes
pizzas
pizza_id TEXT
pizza_type_id TEXT
size TEXT
price DOUBLE
Indexes
orders
order_id INT
order_date DATE
order_time TIME
Indexes

01 Retrieve the total number of orders placed.

```sql
select count(order_id) as total_orders from orders;
```



02 Calculate the total revenue generated from pizza sales.

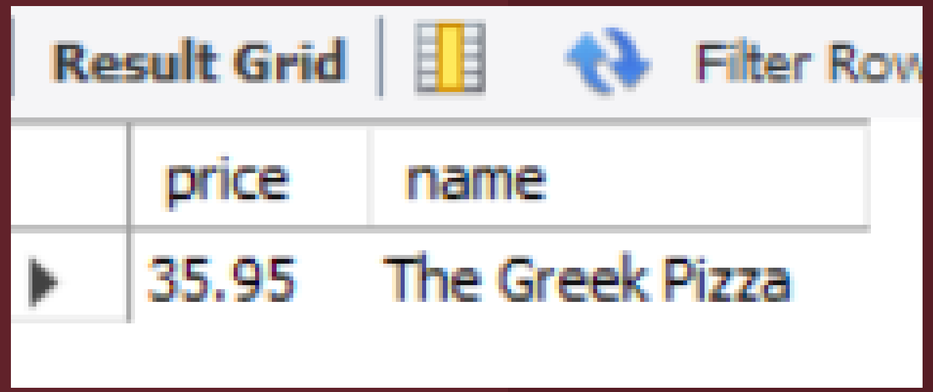
```
SELECT
    ROUND(SUM(pizzas.price * orders_details.quantity),
            2) AS total_revenue
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id
```

Result Grid

total_revenue
817860.05

03 Identify the highest-priced pizza.

```
SELECT
    p.price, pizza_types.name
FROM
    pizza_types
        JOIN
    pizzas p ON pizza_types.pizza_type_id = p.pizza_type_id
ORDER BY p.price DESC
LIMIT 1;
```



04

Identify the most common pizza size ordered.

```
SELECT
    p.size, COUNT(o.quantity) AS count
FROM
    orders_details o
        JOIN
    pizzas p ON o.pizza_id = p.pizza_id
GROUP BY p.size
ORDER BY count DESC
```

Result Grid

size	count
L	18526
M	15385
S	14137
XL	544
XXL	28

05

List the top 5 most ordered pizza types along with their quantities.

```sql
SELECT
    pt.name, SUM(quantity) AS total_quantity
FROM
    orders_details o
        JOIN
    pizzas p ON p.pizza_id = o.pizza_id
        JOIN
    pizza_types pt ON pt.pizza_type_id = p.pizza_type_id
GROUP BY pt.name
ORDER BY total_quantity DESC
LIMIT 5;
```

Result Grid | Filter Rows:

name	total_quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

06 Join the necessary tables to find the total quantity of each pizza category ordered.

```
SELECT
    SUM(o.quantity) AS total_quantity, pt.category
FROM
    orders_details o
        JOIN
    pizzas p ON p.pizza_id = o.pizza_id
        JOIN
    pizza_types pt ON pt.pizza_type_id = p.pizza_type_id
GROUP BY pt.category;
```

Result Grid | Filter Row

total_quantity	category
14888	Classic
11649	Veggie
11987	Supreme
11050	Chicken

07 Determine the distribution of orders by hour of the day.

```sql
SELECT
    HOUR(order_time) AS hourr, COUNT(order_id) as count
FROM
    orders
GROUP BY hourr;
```

Result Grid

hourr	count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399

08 Join relevant tables to find the category-wise distribution of pizzas.

```sql
SELECT
    category, COUNT(name) as count_of_pizza
FROM
    pizza_types
GROUP BY category;
```

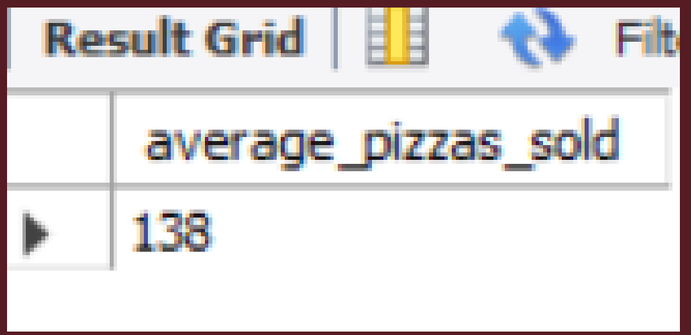
Result Grid | Filter Rows

category	count_of_pizza
Chicken	6
Classic	8
Supreme	9
Veggie	9

09

Group the orders by date and calculate the average number of pizzas ordered per day.

```
SELECT
    ROUND(AVG(quantity), 0) as average_pizzas_sold
FROM
    (SELECT
        order_date, SUM(quantity) AS quantity
    FROM
        orders o
    JOIN orders_details od ON od.order_id = o.order_id
    GROUP BY order_date) AS data
```



10 Determine the top 3 most ordered pizza types based on revenue.

```sql
SELECT
    pt.name, SUM((od.quantity * p.price)) AS revenue
FROM
    orders_details od
        JOIN
    pizzas p ON p.pizza_id = od.pizza_id
        JOIN
    pizza_types pt ON pt.pizza_type_id = p.pizza_type_id
GROUP BY pt.name
ORDER BY revenue DESC
LIMIT 3;
```

Result Grid | Filter Rows:

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

11 Calculate the percentage contribution of each pizza type to total revenue.

```sql
SELECT
    pt.category,
    ROUND((SUM(p.price * od.quantity) / (SELECT
                    SUM(od.quantity * p.price)
                FROM
                    orders_details od
                        JOIN
                    pizzas p ON p.pizza_id = od.pizza_id)) * 100,2) AS revenue
FROM
    pizzas p
        JOIN
    orders_details od ON od.pizza_id = p.pizza_id
        JOIN
    pizza_types pt ON pt.pizza_type_id = p.pizza_type_id
GROUP BY pt.category;
```

category	revenue
Classic	26.91
Veggie	23.68
Supreme	25.46
Chicken	23.96

12

Analyze the cumulative revenue generated over time.

```
select order_date,round(sum(revenue) over(order by order_date),2) as cum_revenue from
(select o.order_date, sum(od.quantity* p.price) as revenue from orders_details od
join  pizzas p on p.pizza_id=od.pizza_id
join orders o on o.order_id = od .order_id group by o.order_date order by o.order_date) as sales
```

Result Grid | Filter Row

order_date	cum_revenue
2015-01-01	2713.85
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7

Result 2

13

Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```
select category,name,revenue from
(select category,name,revenue, rank() over (partition by category order by revenue desc) as rn from
(select pt.category,pt.name,sum((od.quantity*p.price)) as revenue from  orders_details od
join pizzas p on od.pizza_id=p.pizza_id join
pizza_types pt on pt.pizza_type_id = p.pizza_type_id group by pt.category,pt.name)as a) as b where rn<=3 ;
```

Result Grid | Filter Rows:

category	name	revenue
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Classic Deluxe Pizza	38180.5
Classic	The Hawaiian Pizza	32273.25
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75

THANK YOU

02 August, 2024